# Boosting Techniques for Uplift Modelling

Michał Sołtys[1] and Szymon Jaroszewicz[1,2]

[1] Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland
[2] National Institute of Telecommunications
ul. Szachowa 1, 04-894 Warsaw, Poland

**Abstract.** Predicting causal effects of actions taken was always one of the most important aims of human reasoning. Every human action is meant to increase probability of desired circumstances and reduce risks of unwanted ones. Actually, people seem to reason in the following way: if probability of desired outcome after a given action is high enough, it is worth trying. Yet prior chances of success (if action not adopted) are completely ignored, perhaps assumed to be negligibly small.
Unfortunately, this approach suffers from serious drawbacks. Consider for example a typical marketing campaign. Conducted on small random sample of customers, it is used to evaluate a probability of purchase (responding to a campaign) after the action was performed. Then a classification model is built to pick a group of customers, to which the campaign should be addressed. We achieve a model targeting customers most likely to buy *after* the campaign. But this is not what a marketer wants. Some of the customers would have bought regardless of the campaign, targeting them brought unnecessary costs. Other customers were actually going to make a purchase but were annoyed by the campaign. It is a well known phenomenon in the marketing literature; the result is a loss of a sale or even a complete loss of the customer (churn).
We should rather select customers who will buy *because* of the campaign, that is, those who are likely to buy if targeted, but unlikely to buy otherwise. Only then we actually can focus on performing the action to increase our chances, not just act when these chances are relatively high anyway. Notice also that similar problems arise in medicine where some patients may recover without actually being treated and some may be hurt by the therapy's side effects more than by the disease itself.
*Uplift modelling* provides a solution to the described problem. The approach uses two separate training sets: *treatment* and *control*. Individuals in the treatment group are subjected to the action, such as a medical treatment or a marketing campaign. The control dataset contains objects which are not subjected to the action and serve as a background against which its effect can be assessed. Instead of modelling class probabilities, uplift modelling attempts to model the *difference* between conditional class probabilities in the treatment and control groups. This way, the causal influence of the action can be modelled, and the method is able to

predict the true gain (with respect to taking no action) from targeting a given individual.
As the uplift approach is being developed and increasingly appears to be a prospective methodology, the need for more sophisticated tools becomes natural. In the case of classification, apart from more and better algorithms appearing, a hugely important milestone has been the invention of ensemble methods which strengthen existing classification algorithms. This powerful procedures allow to improve performance of many classifiers in a general way, often turning weak single models into highly capable ensembles. It becomes clear then, that a search for an uplift analogue of ensemble methods is needed.
We consider a few methods of applying an idea of the boosting procedure to an uplift approach. These are: a double (classifier) boosting approach being a natural way of implementing uplift boosting; a class variable transformation allowing for application of any ordinary classifiers to uplift modelling, and Uplift AdaBoost being a new algorithm for uplift modelling which realizes one of the basic assumptions of classic boosting: forgetting the last member added to the ensemble in each iteration.
We focus on the mechanism, used in classical boosting, of updating record weights such that its classification error is exactly $1/2$ after each iteration, which makes it likely for the next member to be very different from the previous one, leading to a diverse ensemble.
Implementation of this feature, known as forgetting the last member of the ensemble, is significantly more complex than in classification case. Since we have two datasets, treatment and control, reweighting instances can be done in infinite number of ways. Unlike the classification boosting, we have now two classification accuracies in each iteration, which should be used in establishing model weights; this makes the problem more challenging.
We construct an uplift AdaBoost algorithm preserving the feature of forgetting by setting weight update parameters for treatment and control datasets as well as model weights for each iteration in a way which guarantees convergence. We discuss analogies and dissimilarities between classification and uplift boosting algorithms, including theoretical properties and practical consequences.
We perform an experimental evaluation that demonstrate the usefulness of the methods considered. We compare their performance and performance of the base models on benchmark datasets. A proposed uplift boosting methods often dramatically improve performance of the base models and are thus new and powerful tools for uplift modelling.

## 1 Introduction

The main interest of machine learning is the problem of classification, where the task is to predict, based on a number of attributes, the class to which an instance belongs, or the conditional probability of it belonging to each of the classes. Unfortunately, classification is not well suited to many problems in marketing

or medicine to which it is applied. Let us discuss it on the example of a direct marketing campaign where potential customers receive a mailing offer.

A typical application of machine learning techniques in this context involves selecting a small pilot sample of customers who receive the campaign. Next, a classifier is built based on the pilot campaign outcomes and used to select customers to whom the offer should be mailed. As a result, the customers most likely to buy *after* the campaign will be selected as targets.

Unfortunately this is not what a marketer wants! Some of the customers would have bought regardless of the campaign; targeting them resulted in unnecessary costs. Other customers were actually going to make a purchase but were annoyed by the campaign. The result is a loss of a sale or even a complete loss of the customer (churn). While the second case may seem unlikely, it is a well known phenomenon in the marketing community [1, 2].

In order to run a truly successful campaign, we need, instead, to be able to select customers who will buy *because* of the campaign, i.e., those who are likely to buy if targeted, but unlikely to buy otherwise. Similar problems arise in medicine where some patients may recover without actually being treated and some may be hurt by the therapy's side effects more than by the disease itself.

*Uplift modelling* provides a solution to this problem. The approach employs two separate training sets: *treatment* and *control*. The objects in the treatment dataset have been subject to some action, such as a medical treatment or a marketing campaign. The control dataset contains objects which have not been subject to the action and serve as a background against which its effect can be assessed. Instead of modelling class probabilities, uplift modelling attempts to model the *difference* between conditional class probabilities in the treatment and control groups. This way, the *causal* influence of the action can be modelled, and the method is able to predict the true gain (with respect to taking no action) from targeting a given individual.

While in described problems uplift modelling is a better alternative for standard classification, we should expect a dynamic development of the approach. Yet, despite its practical appeal, uplift modelling has received surprisingly little attention in the literature. There are, however, papers concerning uplift models and successful applications to practical problems, especially in marketing, are reported. An American bank used uplift modelling to turn an unsuccessful mailing campaign into a profitable one [3]. Applications have also been reported in minimizing churn at mobile telecoms [4]. In [5] an approach to online advertising has been proposed which combines uplift modelling with maximizing the response rate in the treatment group to increase advertiser's benefits.

Although there would not be any reservations to use an algorithm to choose who should receive an advert or some marketing campaign, leaving a decision on treatment to some statistical procedure may seem too controversial in medicine. Still, doctors may be interested in factors indiated by the model to be responsible for chances of recovery after the treatment was applied. What is more, uplift modelling allows for any arbitrary number of factors, unlike typical medical trials with control groups.

As uplift approach is developed and seems to be a prospective methodology, a need for more sophisticated tools become natural. As it was in the case of classification, apart from more and better algorithms appearing, there was a marvelous milestone done: ensemble methods were invented to strengthen all existing classification algorithms. This powerful procedures allow to improve performance of any classifier in a generic way, often turning weak single models into highly capable ensembles. It becomes clear then, that search for uplift analogon of ensemble methods is needed.

This paper presents an adaptation of AdaBoost algorithm to the uplift modelling case. Boosting often dramatically improves performance of classification models, and in this paper we demonstrate that it can bring similar benefits to uplift modelling. We apply forgetting the last member of the ensemble to the described problem, trying to repeat the success of the classical algorithm in the uplift case. Experimental verification proves that the benefits of boosting extend to the case of uplift modelling and shows relative merits of the new approach.

In the remaining part of this section we introduce a definition of an uplift analogue of classification error and present two alternative ways to apply boosting procedures to the uplift case: a class variable transformation and a double classifier approach. We give an overview of the other related work and remind the property of forgetting the last member of the ensemble in classification boosting. But first we have to start with introducing a notation used throughout the paper.

### 1.1 Notation

We will now introduce the notation used further in the article. We use the superscript $^T$ for quantities related to the treatment group and the superscript $^C$ for quantities related to the control group. For example, the treatment training dataset will be denoted with $\mathcal{D}^T$ and the control training dataset with $\mathcal{D}^C$. Both datasets together constitute the whole training dataset, $\mathcal{D} = \mathcal{D}^T \cup \mathcal{D}^C$.

Each data record $(x, y)$ consists of a vector of features $x \in \mathcal{X}$ and a class $y \in \{0, 1\}$ with 1 assumed to be the successful outcome, for example patient recovery or a positive response to a marketing campaign. Let $N^T$ and $N^C$ denote the number of records in the treatment and control datasets.

An uplift model is a function $h : \mathcal{X} \to \{0, 1\}$. The value $h(x) = 1$ means the action is deemed beneficial for $x$ by the model, $h(x) = 0$ means that its impact is considered neutral or negative. By 'positive outcome' we mean that the probability of success for a given individual $x$ is higher if the action is performed on her than if the action is not taken.

We will denote general probabilities related to the treatment and control groups with $P^T$ and $P^C$, respectively. For example, $P^T(y = 1, h = 1)$ stands for probability that a randomly selected case in the treatment set has a positive outcome and taking the action on it is predicted to be beneficial by an uplift model $h$. We can now state more formally when an individual $x$ should be subject to an action, namely, when $P^T(y = 1|x) - P^C(y = 1|x) > 0$.

In the $m$-th step of the boosting algorithm the $i$-th treatment group training record is assumed to have a weight $w^T_{m,i}$ assigned to it. Likewise a weight $w^C_{m,i}$ is assigned to the $i$-th control training case. Further, denote by

$$p^T_m = \frac{\sum_{i=1}^{N^T} w^T_{m,i}}{\sum_{i=1}^{N^T} w^T_{m,i} + \sum_{i=1}^{N^C} w^C_{m,i}}, \qquad p^C_m = \frac{\sum_{i=1}^{N^C} w^C_{m,i}}{\sum_{i=1}^{N^T} w^T_{m,i} + \sum_{i=1}^{N^C} w^C_{m,i}} \tag{1}$$

the relative sizes of treatment and control datasets at step $m$. Notice that $p^T_m + p^C_m = 1$ for every $m$.

### 1.2 An uplift analogue of classification error

We begin with mentioning a problem which is the biggest challenge of uplift modelling as opposed to standard classification. The problem has been known in statistical literature (see [6]) as the

> **Fundamental Problem of Causal Inference.** For every individual, only one of the outcomes is observed, after the individual has been subject to an action (treated) or when the individual has not been subject to the action (was a control case), *never* both.

As a result we never know whether the action performed on a given individual was truly beneficial. This is different from classification, where the true class of each individual in the training set is known.

Due to the Fundamental Problem of Causal Inference we cannot tell whether an uplift model correctly classified a given instance. We will, however, define an approximate notion of classification error in the uplift case. A record $(x^T_i, y^T_i)$ is assumed to be classified correctly by an uplift model $h$ if $h(x^T_i) = y^T_i$ and $(x^T_i, y^T_i) \in \mathcal{D}^T$; a record $(x^C_i, y^C_i)$ is assumed to be classified correctly if $h(x^C_i) = 1 - y^C_i$ and $(x^C_i, y^C_i) \in \mathcal{D}^C$.

Intuitively, if a record $(x^T_i, y^T_i)$ belongs to the treatment group and a model $h$ predicts that it should receive the treatment ($h(x^T_i) = 1$) then the outcome should be positive ($y^T_i = 1$) if the recommendation is to be correct. Note that the gain from the action might also be neutral if a success would have occurred also without treatment, but at least the model's recommendation is not in contradiction with the observed outcome. If, on the contrary, the outcome for a record in the treatment group is 0 and $h(x^T_i) = 1$, the prediction is clearly wrong as the true effect of the action can at best be neutral.

In the control group the situation is reversed. If the outcome was positive ($y^C_i = 1$) but the model predicted that the treatment should be applied ($h(x^C_i) = 1$), the prediction is clearly wrong, since the treatment cannot be truly beneficial, it can at best be neutral. To simplify notation we will introduce

the following indicators:

$$e^T(x_i^T) = \begin{cases} 0 & \text{if } x_i^T \in \mathcal{D}^T \text{ and } h(x_i^T) = y_i^T, \\ 1 & \text{if } x_i^T \in \mathcal{D}^T \text{ and } h(x_i^T) \neq y_i^T, \end{cases} \tag{2}$$

$$e^C(x_i^C) = \begin{cases} 0 & \text{if } x_i^C \in \mathcal{D}^C \text{ and } h(x_i^C) \neq y_i^C, \\ 1 & \text{if } x_i^C \in \mathcal{D}^C \text{ and } h(x_i^C) = y_i^C. \end{cases} \tag{3}$$

An index $m$ will be added to indicate the $m$-th step of the algorithm. Let us now define uplift analogues of classification error on the treatment and control datasets and a combined error:

$$\epsilon_m^T = \frac{\sum_{i:\ e_m^T(x_i)=1} w_{m,i}^T}{\sum_{i=1}^{N^T} w_{m,i}^T}, \quad \epsilon_m^C = \frac{\sum_{i:\ e_m^C(x_i)=1} w_{m,i}^C}{\sum_{i=1}^{N^C} w_{m,i}^C}, \quad \epsilon_m = p_m^T \epsilon_m^T + p_m^C \epsilon_m^C. \tag{4}$$

The sums above are a shorthand notation for summing over misclassified instances in the treatment and control training sets, which will also be used later in the paper.

### 1.3 Double classifiers

The most obvious approach to uplift modelling is to build two classification models $h^T$ and $h^C$ on the treatment and control groups respectively and to subtract their predicted probabilities:

$$h^U(\mathbf{x}) = h^T(\mathbf{x}) - h^C(\mathbf{x}).$$

We will call this approach the *double classifier* approach. Its obvious appeal is simplicity; however in many cases the approach may perform poorly. The reason is that both models can focus on predicting the class probabilities themselves, instead of making the best effort to predict the (usually much weaker) 'uplift signal', i.e., the difference between conditional class probabilities in the treatment and control groups. See [2] for a detailed discussion and an illustrative example[1]. Nevertheless, in some cases the approach is competitive. This is the case when the amount of training data is large enough to accurately estimate conditional class probabilities in both groups or when the net gain is correlated with the class variable, e.g. when people likely to buy a product are also likely to positively respond to a marketing offer related to that product.

### 1.4 Class variable transformation

In [7] a class variable transformation was presented which allows for converting an arbitrary classification model (the paper used logistic regression) into an

[1] The example is based on artificial data with two attributes, one strongly affecting the class probabilities independently from the treatment received, the other determining the relatively small sensitivity to the treatment. A model based on two decision trees uses only the first attribute.

uplift model. The transformation simply replaces class values $y_i^C$ in the control group with their reverses $1 - y_i^C$ while keeping the treatment set class values unchanged. As a result, a single classifier is built which directly models the difference between success probabilities in the treatment and control groups. It is easy to see that the errors defined in Equation 4 are equivalent to standard classification errors for the transformed class.

### 1.5 Other related work

Despite its practical appeal, uplift modelling has seen relatively little attention in the literature. Here we shortly discuss some other work not mentioned above.

Several algorithms have thus been proposed which directly model the difference between class probabilities in the treatment and control groups. Many of them are based on modified decision trees. For example, [2] describe an uplift tree learning algorithm which selects splits based on a statistical test of differences between treatment and control class probabilities. In [8, 9] uplift decision trees based on information theoretical split criteria have been proposed.

Some work has also been published on using ensemble methods for uplift modelling, although, to the best of our knowledge, none of them on boosting. Bagging of uplift models has been mentioned in [2]. Uplift Random Forests have been proposed by [10]; an extension, called causal conditional inference trees was proposed by the same authors in [11]. A thorough experimental and theoretical analysis of bagging and random forests in uplift modelling can be found in [12] where it is argued that ensemble methods are especially well suited to this task and that bagging performs surprisingly well.

Other uplift techniques have also been proposed. Regression based approaches can be found in [13] or, in a medical context, in [14, 15].

[16] proposes a method for converting survival data such that uplift modelling can, under certain assumptions, be directly applied to it.

Some variations on the uplift modelling theme have also been explored. [5] proposed an approach in the context of online advertising, where it is necessary to not only maximize the net gain, but also to increase advertiser's benefits through maximizing response rate in the treatment group. This type of problems are beyond the scope of this paper.

### 1.6 Forgetting in classical AdaBoost

While many boosting algorithms are available, in this paper by 'boosting' we mean the discrete AdaBoost algorithm [17]. Forgetting the last member added to the ensemble means that after a new member is added, record weights are updated such that its classification error is exactly 1/2. This makes it likely for the next member to be very different from the previous one, leading to a diverse ensemble. Full details can be found for example in [17–19]. This key property will be important for adapting boosting to the uplift modelling case.

Now we can formulate an uplift analogon of AdaBoost algorithm.

## 2 Uplift AdaBoost

In this section we present the proposed algorithm and the property of forgetting the last ensemble member in the context of uplift modelling.

### 2.1 Algorithm

Algorithm 1 presents AdaBoost algorithm for uplift modelling.

**Input:** set of treatment training records, $\mathcal{D}^T = \{(x_1^T, y_1^T), \ldots, (x_{N^T}^T, y_{N^T}^T)\}$,
set of control training records, $\mathcal{D}^C = \{(x_1^C, y_1^C), \ldots, (x_{N^C}^C, y_{N^C}^C)\}$,
base uplift algorithm to be boosted,
integer $M$ specifying the number of iterations

1. Initialize weights $w_{1,i}^T, w_{1,i}^C$
2. For $m \leftarrow 1, \ldots, M$
   (a) $w_{m,i}^T \leftarrow \frac{w_{m,i}^T}{\sum_j w_{m,j}^T + \sum_j w_{m,j}^C}$; $w_{m,i}^C \leftarrow \frac{w_{m,i}^C}{\sum_j w_{m,j}^T + \sum_j w_{m,j}^C}$
   (b) Build a base model $h_m$ on $\mathcal{D}$ with $w_{m,i}^T, w_{m,i}^C$
   (c) Compute the treatment and control errors $\epsilon_m^T$, $\epsilon_m^C$
   (d) Compute $\beta_m = \frac{p_m^T \epsilon_m^T + p_m^C \epsilon_m^C}{1 - p_m^T \epsilon_m^T - p_m^C \epsilon_m^C}$
   (e) If $\beta_m = 1$ or $\epsilon_m^T \notin (0, \frac{1}{2})$ or $\epsilon_m^C \notin (0, \frac{1}{2})$:
      i. choose random weights $w_{m,i}^T$, $w_{m,i}^C$
      ii. continue with next boosting iteration
   (f) $w_{m+1,i}^T \leftarrow w_{m,i}^T \cdot (\beta_m)^{1[h_m(x_i^T) = y_i^T]}$
   (g) $w_{m+1,i}^C \leftarrow w_{m,i}^C \cdot (\beta_m)^{1[h_m(x_i^C) = 1 - y_i^C]}$
   (h) Add $h_m$ with coefficient $\beta_m$ to the ensemble

**Output:** The final hypothesis

$$h_f(x) = \begin{cases} 1 \; if \; \sum_{m=1}^M \left(\log \frac{1}{\beta_m}\right) h_m(x) \geq \frac{1}{2} \sum_{m=1}^M \log \frac{1}{\beta_m}, \\ 0 \; otherwise. \end{cases} \tag{5}$$

**Algorithm 1:** AdaBoost algorithm for uplift.

Note that the algorithm is a discrete boosting algorithm [17, 19], that is, the base learners are assumed to return a discrete decision on whether the action should be taken (1) or not (0). Algorithm 1, as presented in the figure, also returns a decision. However, it can also return a numerical score,

$$s(x) = \sum_{m=1}^{M} \left(\log \frac{1}{\beta_m}\right) h_m(x),$$

indicating how likely it is that the effect of the action is positive on a given case. In the experimental Section 3 we will use this variant of the algorithm.

AdaBoost can suffer from premature stops when the sum of weights of misclassified cases becomes 0 or is greater than 1/2. This problem turns to be even more troublesome in the uplift modelling case. Hence, in step 2e of Algorithm 1 we restart the algorithm by assigning random weights drawn from the exponential distribution to records in both training datasets. The technique has been suggested for classification boosting in [20].

### 2.2 Properties

Let us now examine what the property of forgetting the last model added to the ensemble means in the context of uplift error defined in Equation 4. To forget the member $h_m$ added in step $m$ we need to choose weights in step $m+1$ such that the combined error of $h_m$ is exactly one half, $\epsilon_m = \frac{1}{2}$. From steps 2f and 2g of Algorithm 1 we get that to ensure the condition holds at step $m+1$, the following equation for $\beta_m$ must be true:

$$\beta_m \sum_{i:\, e_m^T(x_i)=0} w_{m,i}^T + \beta_m \sum_{i:\, e_m^C(x_i)=0} w_{m,i}^C = \sum_{i:\, e_m^T(x_i)=1} w_{m,i}^T + \sum_{i:\, e_m^C(x_i)=1} w_{m,i}^C, \quad (6)$$

that is, the total new weights of correctly classified examples need to be equal to total new weights of incorrectly classified examples. After dividing both sides by $\sum_{i=1}^{N^T} w_{m,i}^T + \sum_{i=1}^{N^C} w_{m,i}^C$ the equation becomes

$$p_m^T(1-\epsilon_m^T)\beta_m + p_m^C(1-\epsilon_m^C)\beta_m = p_m^T\epsilon_m^T + p_m^C\epsilon_m^C. \quad (7)$$

Note that unlike classical boosting, this condition does not uniquely determine record weights.

Let us now give a justification of this condition in terms of performance of an uplift model.

**Theorem 1.** *Let $h$ be an uplift model. If the balance condition holds and the assignment of cases to the treatment and control groups is random then the condition that the combined uplift error $\epsilon$ be equal to $\frac{1}{2}$ is equivalent to*

$$P(h=1)\left[P^T(y=1|h=1) - P^C(y=1|h=1)\right] + P(h=0)\left[P^C(y=1|h=0) - P^T(y=1|h=0)\right] = 0. \quad (8)$$

*Proof.* Note that the assumption of random group assignment implies $P^T(h=1) = P^C(h=1) = P(h=1)$ since both groups are scored with the same model and have the same distributions of predictor variables. Using the balance condition, the error $\epsilon$ of $h$, defined in Equation 4, can be expressed as (the second equality follows from $p^T = p^C = \frac{1}{2}$)

$$\begin{aligned} 2\epsilon &= 2P^T(h=1-y)p^T + 2P^C(h=y)p^C = P^T(h=1-y) + P^C(h=y) \\ &= P^T(h=1,y=0) + P^T(h=0,y=1) + P^C(h=y=0) + P^C(h=y=1) \\ &= P^T(y=0|h=1)P^T(h=1) + P^T(y=1|h=0)P^T(h=0) \\ &\quad + P^C(y=1|h=1)P^C(h=1) + P^C(y=0|h=0)P^C(h=0). \end{aligned}$$

Using the assumption of random treatment assignment and rearranging:

$$\begin{aligned}
&= P(h=1)\left[P^T(y=0|h=1) + P^C(y=1|h=1)\right] \\
&\quad + P(h=0)\left[P^T(y=1|h=0) + P^C(y=0|h=0)\right] \\
&= P(h=1)\left[1 - P^T(y=1|h=1) + P^C(y=1|h=1)\right] \\
&\quad + P(h=0)\left[P^T(y=1|h=0) + 1 - P^C(y=1|h=0)\right] \\
&= 1 + P(h=1)\left[-(P^T(y=1|h=1) - P^C(y=1|h=1))\right] \\
&\quad + P(h=0)\left[P^T(y=1|h=0) - P^C(y=1|h=0)\right].
\end{aligned}$$

After taking $\epsilon = \frac{1}{2}$ the result follows.

Note that the left term in (8) is the total gain in success probability due to the action being taken on cases selected by the model and the right term is the gain from not taking the action on cases not selected by the model. A good uplift model tries to maximize both quantities, so the sum being equal to zero corresponds to a model giving no overall gain over the controls.

When the balance condition holds, the forgetting property thus has a clear interpretation in terms of uplift model performance. When the balance condition does not hold, the interpretation is, at least partially, lost.

Note that $\beta_m$ we choose:

$$\beta_m = \frac{p_m^T \epsilon_m^T + p_m^C \epsilon_m^C}{1 - (p_m^T \epsilon_m^T + p_m^C \epsilon_m^C)} \tag{9}$$

is identical to the result in classical boosting with the classification error being replaced by its uplift analogue.

## 3 Experimental evaluation

In this section we present an experimental evaluation of the three proposed algorithms and compare their performance with performance of the base models. We begin by describing the test datasets we are going to use, then review the approaches to evaluating uplift models and finally present the experimental results.

### 3.1 Benchmark datasets

A significant problem one encounters while working on uplift modelling is the lack of publicly available datasets. Even though control groups are ubiquitous in medicine and their use in marketing is growing, there are relatively few publicly available datasets which include a control group and a reasonable number of predictive attributes. In our experiments we are going to use datasets from the UCI repository artificially split into treatment and control groups. We describe

**Table 1.** Conversion of UCI datasets into treatment and control groups.

| dataset | treatment/control split condition | #removed attributes / # original attributes |
|---|---|---|
| breast-cancer | menopause = 'PREMENO' | 2/9 |
| credit-a | a7 $\neq$ 'V' | 3/15 |
| dermatology | exocytosis $\leq 1$ | 16/34 |
| liver-disorders | drinks $< 2$ | 2/6 |
| splice | attribute1 $\in \{$'A', 'G'$\}$ | 2/61 |
| winequal-red | sulfur dioxide $< 46.47$ | 2/11 |

here the procedure used to split standard UCI datasets in a way suitable for uplift modelling. The details of the approach can be found in [8, 9].

The conversion is performed by first picking one of the data attributes which splits the data evenly into two groups. Details are given in Table 1. The first column contains the dataset name and the second provides the condition used to select records for the treatment group. The remaining records formed the control. A further postprocessing step removed attributes strongly correlated with the split itself; ideally, the division into treatment and control groups should be independent from all predictive attributes, but this is possible only in a controlled experiment. A simple heuristic was used for this purpose:

1. A numerical attribute was removed if its means in the treatment and control datasets differed by more than 25%.
2. A categorical attribute was removed if the probability of one of its categories differed between the treatment and control datasets by more than 0.25.

The number of removed attributes vs. the total number of attributes is shown in the third column of Table 1.

Further, multiclass problems were converted into binary problems with the majority class assumed to be class 1 (the desired outcome) and the remaining classes merged into class 0. We note that it is possible to use all analyzed uplift methods in the multiclass setting, however, we chose to use binarization in order to make the analysis (e.g. drawing curves) easier.

### 3.2 Methodology

Building uplift models requires two training sets. Consequently, we also have two test sets: treatment and control. A typical approach to assessing uplift models [2, 1] is to score both test datasets using the same uplift model and assume that objects in the treatment and control groups which have received similar scores are similar and can be compared with each other. In [1] the authors grouped treatment and control test cases by deciles of their scores and estimated net gains by subtracting success rates within each decile.

A more practical modification of this approach is to visualize model performance using *uplift curves* [8, 2]. Recall that one of the tools for assessing performance of standard classification models are lift curves[2], where the $x$ axis corresponds to the number of cases subjected to an action and the $y$ axis to the number of successes captured by the model.

In order to obtain an *uplift curve* we score both test sets using the uplift model and subtract the lift curve generated on the control test set from the lift curve generated on the treatment test set. The number of successes for both curves is expressed as percentage of the total population such that the subtraction is meaningful.

The interpretation of the uplift curve is as follows: on the $x$ axis we select the percentage of the population on which the action is performed, and on the $y$ axis we read the net gain achieved on the targeted group (the net gain on the remaining cases is zero since no action was performed on them). The point at $x = 100\%$ gives the gain in success probability we would obtain if the action was applied to the whole population. A diagonal uplift curve corresponds to performing the action on a randomly selected percentage of the population. More details can be found in [8, 2].

As with ROC curves, we can use the Area Under the Uplift Curve (AUUC) to summarize model performance with a single number. We subtract the area under the diagonal from this value in order to obtain more meaningful numbers. Note that the area under the uplift curve can be less than zero; this happens when the model gives high scores to cases for which the action has a predominantly negative effect.

All experiments have been performed by randomly splitting each dataset into training (80% of the data) and test (the remaining 20%) parts. Each experiment was repeated 128 times, and the resulting uplift curves have been averaged. The reason for this choice was to make the results repeatable and less sensitive to the random seed used. However, the disadvantage of such an approach is that it hides the variance of the predictions. To address this issue we also compute standard deviations of AUUCs computed over the 128 test sets in a manner similar to bootstrap estimates.

### 3.3 Experiments

As base models to be boosted we use two types of decision trees: unpruned J4.8 trees and decision stumps implemented in `Weka` package. We apply to them the three methods of boosting in the uplift approach: double (classical) boosting, class variable transformation and uplift AdaBoost algorithm proposed by us.

Thus we obtain two base models:

- a double classifier,
- a classifier with the class variable transformation

and four boosted models:

[2] Also known as cumulative gains curves or cumulative accuracy profiles.

- a doubled classical boosting ensemble,
- uplift AdaBoost ensemble with doubled classifiers,
- a classical boosting ensemble of classifiers with class variable transformation,
- uplift AdaBoost ensemble of classifiers with class variable transformation.

The class variable transformation is named shortly *Z model*, e.g. a decision tree with class variable transformation is named "Z decision tree". Note that "doubling" and "Z transformation" are two different ways of achieving uplift models, which than can be boosted with Uplift AdaBoost. Alternatively, we can double (classically) boosted classifiers or classically boost Z models.

In each ensemble we build $B = 101$ base models being members of the ensemble. This choice of the ensemble size is justified by the trade off: the $B$ large enough to get a fully developed ensemble and not too big for practical applications.

Figures 1 to 6 present the uplift curves for chosen UCI datasets and the algorithms applied to J4.8 unpruned decision tree as a base model. In most cases boosting generally improves the base double model and often the proposed uplift model is superior to the ordinary double boosted model. In some cases the latter can eventually fail, which did not happen with the new algorithm (see Figure 2). Note also that the class variable transformation usually does not work properly with uplift AdaBoost.

For decision stumps the results are not so impressive. In fact, this base model sometimes works fine with classical boosting on the data with the class variable transormation, but not for the uplift AdaBoost with variable transformation (not presented on Figures).

## 4 Conclusions

In this paper we have developed a new boosting algorithm for the uplift modelling problem. We discuss some of its properties in relation to the classification AdaBoost algorithm and present the two other approaches to boosting in the uplift case.

Experimental evaluation showed that boosting has a potential to dramatically improve the performance of uplift models and the proposed algorithm often outperform the other two approaches. Our experiments demonstrate that ensemble methods often bring dramatic improvements in performance, turning useless single trees into highly capable ensembles. In some cases the Area Under the Uplift Curve of an ensemble was over double that of the base learner.

We conclude that further investigation of the designed algorithm is very promising and should be continued for various types of base models, as for some of them a possible improvement of model accuracy may be very remarkable.

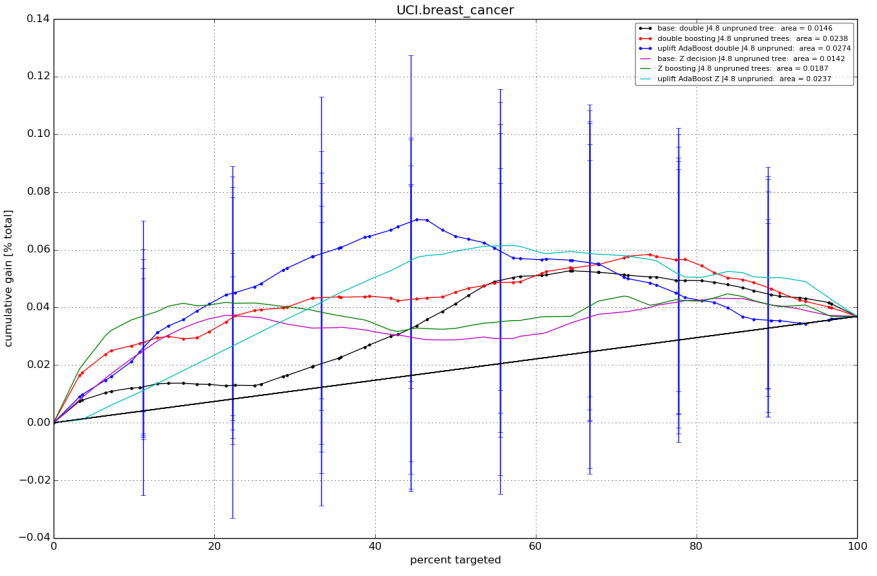


**Fig. 1.** Uplift curves for breast-cancer dataset.

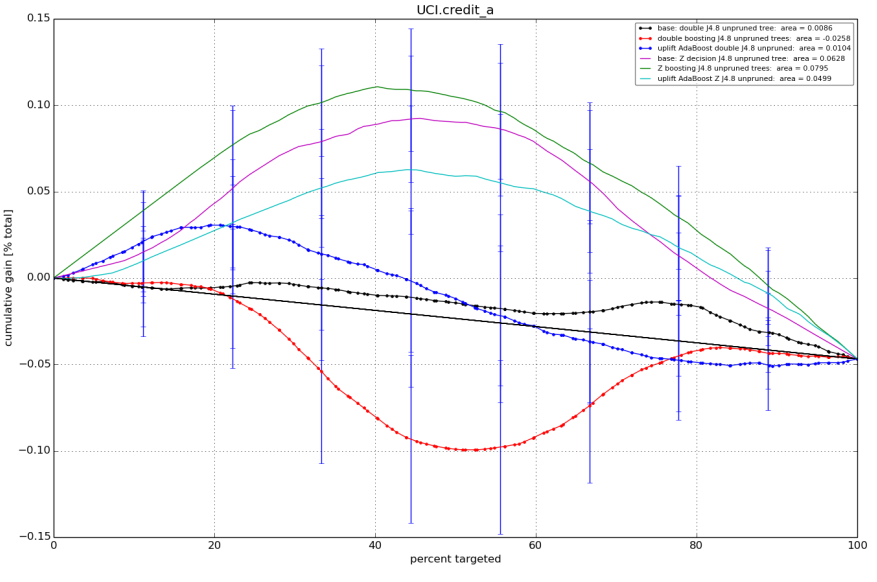


**Fig. 2.** Uplift curves for credit-a dataset.

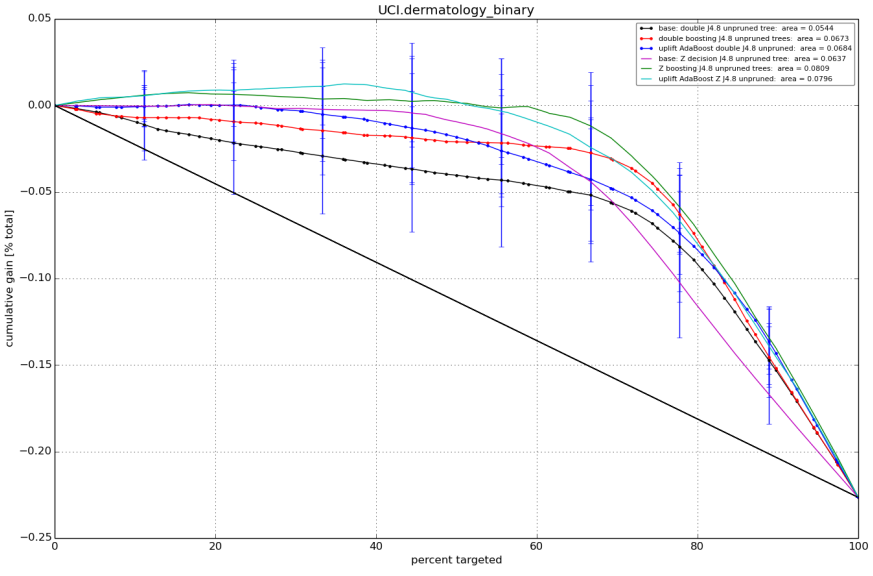


**Fig. 3.** Uplift curves for dermatology dataset.

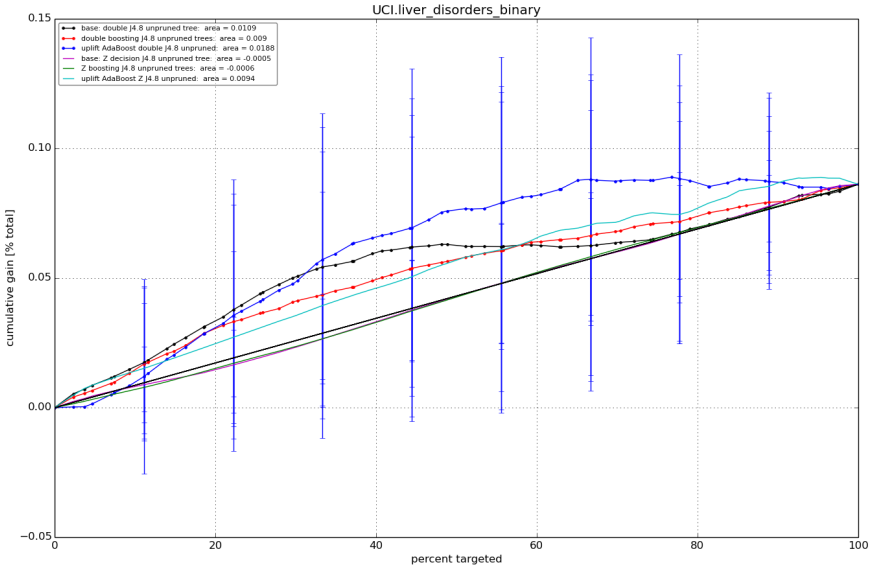


**Fig. 4.** Uplift curves for liver-disorders dataset.

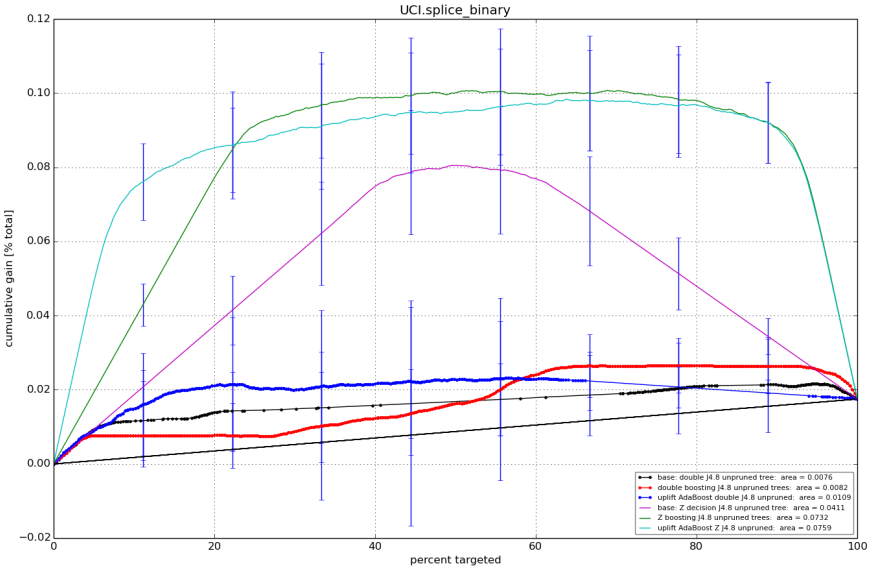


**Fig. 5.** Uplift curves for splice dataset.

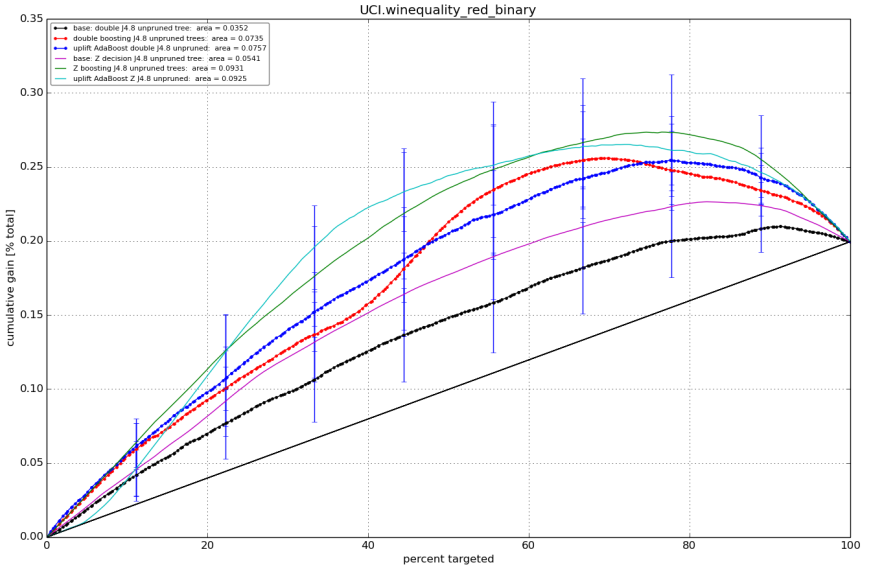


**Fig. 6.** Uplift curves for winequal-red dataset.

## Acknowledgements

This work was supported by Research Grant no. N N516 414938 of the Polish Ministry of Science and Higher Education (Ministerstwo Nauki i Szkolnictwa Wyższego) from research funds for the period 2010–2014. M.S. was also supported by the European Union from resources of the European Social Fund: Project POKL 'Information technologies: Research and their interdisciplinary applications', Agreement UDA-POKL.04.01.01-00-051/10-00.

## References

1. Hansotia, B., Rukstales, B.: Incremental value modeling. Journal of Interactive Marketing **16**(3) (2002) 35–46
2. Radcliffe, N., Surry, P.: Real-world uplift modelling with significance-based uplift trees. Portrait Technical Report TR-2011-1, Stochastic Solutions (2011)
3. Grundhoefer, M.: Raising the bar in cross-sell marketing with uplift modeling. Predictive Analytics World Conference (2009)
4. Radcliffe, N., Simpson, R.: Identifying who can be saved and who will be driven away by retention activity. Journal of Telecommunications Management **1**(2) (April 2008) 168
5. Pechyony, D., Jones, R., Li, X.: A joint optimization of incrementality and revenue to satisfy both advertiser and publisher. In: WWW 2013 Companion. (2013)
6. Holland, P.: Statistics and causal inference. Journal of the American Statistical Association **81**(396) (December 1986) 945–960
7. Jaśkowski, M., Jaroszewicz, S.: Uplift modeling for clinical trial data. In: ICML 2012 Workshop on Machine Learning for Clinical Data Analysis, Edinburgh, Scotland (June 2012)
8. Rzepakowski, P., Jaroszewicz, S.: Decision trees for uplift modeling. In: Proc. of the 10th IEEE International Conference on Data Mining (ICDM), Sydney, Australia (December 2010) 441–450
9. Rzepakowski, P., Jaroszewicz, S.: Decision trees for uplift modeling with single and multiple treatments. Knowledge and Information Systems **32** (August 2012) 303–327
10. Guelman, L., Guillén, M., Pérez-Marín, A.: Random forests for uplift modeling: An insurance customer retention case. In: Modeling and Simulation in Engineering, Economics and Management. Volume 115 of Lecture Notes in Business Information Processing (LNBIP). Springer (2012) 123–133
11. Guelman, L., Guillén, M., Pérez-Marín, A.: A survey of personalized treatment models for pricing strategies in insurance. Insurance: Mathematics and Economics (2014) to appear.
12. Sołtys, M., Jaroszewicz, S., Rzepakowski, P.: Ensemble methods for uplift modeling. Data Mining and Knowledge Discovery (2014) 1–29 online first.
13. Lo, V.: The true lift model - a novel data mining approach to response modeling in database marketing. SIGKDD Explorations **4**(2) (2002) 78–86
14. Robins, J., Rotnitzky, A.: Estimation of treatment effects in randomised trials with non-compliance and a dichotomous outcome using structural mean models. Biometrika **91**(4) (2004) 763–783
15. Vansteelandt, S., Goetghebeur, E.: Causal inference with generalized structural mean models. Journal of the Royal Statistical Society B **65**(4) (2003) 817—835

16. Jaroszewicz, S., Rzepakowski, P.: Uplift modeling with survival data. In: ACM SIGKDD Workshop on Health Informatics (HI-KDD'14), New York City, USA (August 2014)
17. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1) (1997) 119–139
18. Schapire, R.: The strength of weak learnability. Machine Learning **5**(2) (July 1990) 197–227
19. Schapire, R., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine learning **37**(3) (1999) 297–336
20. Webb, G.I.: Multiboosting: A technique for combining boosting and wagging. Machine Learning **40** (2000) 159–196